Довідка AppleScript

Основи

Про AppleScript

Під час користування комп’ютером деякі завдання потрібно виконувати знову й знову. За допомогою AppleScript можна створювати скрипти та програми, які можуть автоматизувати виконання повторюваних завдань.

AppleScript — потужна та гнучка мова скриптування, вбудована до операційної системи OS X. AppleScript можна використовувати для створення клавіатурних скорочень, автоматизації повторюваних завдань або навіть для створення користувацьких програм, які допоможуть заощадити ваш час. Мова AppleScript дозволяє контролювати програми, що підтримують скриптування.

Мова AppleScript складається з команд, схожих на англійську мову, які виконують дії з об’єктами. Прикладами дій є open (відкрити), close (закрити) або get the name of (отримати назву). Прикладами об’єктів є програма, вікно або значення параметра.

Скрипт може керувати однією чи кількома програмами. Для підтримки скриптування програма має визначити терміни, які можна використовувати у скриптах. Операційна система OS X постачається з кількома програмами, які підтримують скриптування.

Скрипти можуть також керувати програмами через мережу, тому можна автоматизувати завдання одразу на кількох комп’ютерах.

Фактично, ви вже використовуєте AppleScript, не усвідомлюючи цього. Більшість функцій системи і програм насправді є скриптами AppleScript, які викликаються з програми. Наприклад, у цій довідковій системі в разі натиснення посилання, яке відкриває програму або файл, виконується відповідний скрипт AppleScript.

ПОВ’ЯЗАНІ ТЕМИ

Як працює AppleScript
Про Редактор AppleScript
Про мову AppleScript
Інтернет-ресурси AppleScript
Розробницьке середовище AppleScript Studio

Основи

Про Редактор AppleScript

Редактор AppleScript — це проста у використанні програма для роботи зі скриптами. Вона може читати, писати, записувати та зберігати скрипти AppleScript.

До функцій Редактора AppleScript належать:

- - забезпечення уніфікованого, простого у використанні інтерфейсу;
- - можливість пошуку й заміни елементів тексту скриптів;
- - автоматичне перенесення довгих операторів скриптів;

- - можливість відміяти дії кілька разів підряд;
- - контекстне меню скриптів для їх швидшого написання;
- - підтримка служб програми;
- - доступ до скриптів програми за допомогою функції «Палітри бібліотеки»;
- - наявність Помічника з написання скриптів, який допомагає завершувати оператори;
- - наявність журналу результатів і журналу реєстрації подій, які дають можливість перевірити перебіг виконання скрипту.

ПОВ'ЯЗАНІ ТЕМИ

Про мову AppleScript
Розробницьке середовище AppleScript Studio

Основи

Створення скриптів AppleScript

Щоб успішно користуватись програмою AppleScript, не потрібно щось знати про програмування. Насправді, можна не писати жодного слова скрипту. До операційної системи OS X входять сотні корисних скриптів AppleScript, які доступні для використання. А ще тисячі скриптів можна знайти в Інтернеті.

Якщо потрібно зробити щось складніше, можна взяти наявний скрипт і пристосувати його до своїх потреб із мінімальними змінами.

Звичайно, найкращого ефекту можна досягти, створюючи власні скрипти. Коли ви навчитесь писати скрипти, то зможете створювати корисні інструменти та автоматизувати виконання завдань для підвищення ефективності своєї роботи.

За допомогою Редактора AppleScript можна створювати власні скрипти та змінювати інші. Він пропонує просте середовище, в якому можна створювати скрипти. Усі приклади в довідці AppleScript створені за допомогою Редактора AppleScript.

- Відкрийте Редактор AppleScript.

  Введіть наведений нижче текст у вікні Редактора AppleScript.

  ```
  say "Welcome to AppleScript!"
  ```

  Натисніть кнопку «Виконати».

ПОВ'ЯЗАНІ ТЕМИ

Про Редактор AppleScript
Розробницьке середовище AppleScript Studio

Основи

Як працює AppleScript

AppleScript працює шляхом надсилання повідомлень, які називаються «події Apple». Коли ви запускаєте скрипт, AppleScript відправляє події Apple до системи або програм, які виконують команди.

У більшості випадків програми або система повертають до AppleScript результат. Наприклад, якщо скрипт надсилає запит програмі на отримання назви об'єкта, програма повертає назву. Якщо команду не можна виконати, тоді програма або AppleScript можуть повертати повідомлення про помилку. Повернені результати можна переглянути в Редакторі AppleScript.

Загалом, не потрібно знати механіку роботи подій Apple. AppleScript здійснює все це у фоновому режимі. Вам потрібно лише писати скрипти та запускати їх.

ПОВ'ЯЗАНІ ТЕМИ

Про AppleScript
Про мову AppleScript

Основи

Про аплети

Скрипт AppleScript, збережений у вигляді програми, називається програмою AppleScript, або аплетом. Аплети автоматично запускаються в разі відкривання та виконують команди в скрипті. Аплети можуть бути маленькими функціями, що виконують лише кілька команд, або складними програмами, що містять тисячі рядків коду.

Аплети можна зберігати як змінювані файли або як виконувані файли. Змінюваний аплет містить текст скрипту, тому його можна редагувати та знову компілювати. Щоб редагувати або змінювати аплет, перетягніть його на іконку Редактора AppleScript. Коли скрипт зберігається як виконуваний файл, редагована частина тексту вилучається. Відтак файл більше не можна редагувати або знову компілювати.

Щоб редагувати аплет, який збережено як змінюваний файл, його потрібно відкрити в Редакторі AppleScript або в іншій програмі для редагування скриптів AppleScript. Аплети вимагають наявності обробника on run, який починає виконувати скрипт під час відкривання аплета.

- У наведеному нижче прикладі показано застосування обробника on run.

```
on run
        --first command
        --second command
end run
```

ПОВ'ЯЗАНІ ТЕМИ

Про дроплети
Типи файлів AppleScript

Основи

Про скомпільовані скрипти

Скомпільований скрипт — це файл AppleScript у доступному для редагування вигляді, який можна читати та редагувати в Редакторі AppleScript або в іншій програмі для редагування скриптів AppleScript.

Коли ви відкриваєте скомпільований скрипт, він відкривається в Редакторі AppleScript, але команди скрипту автоматично не виконуються. Це дозволяє змінювати скрипт, не запускаючи його. Потім можна скомпілювати скрипт, що призведе до перетворення тексту на код AppleScript у файлі. Скомпільований скрипт можна запускати. Файли скомпільованих скриптів мають розширення .scpt.

Коли файл зберігається як скомпільований скрипт, текст залишається у файлі разом з кодом. Тому скрипт можна знову редагувати за потреби.

ПОВ'ЯЗАНІ ТЕМИ

Про аплети
Про дроплети
Типи файлів AppleScript

Основи

Про дроплети

Дроплет — це скрипт AppleScript, який виконується за перетягування файлів або папок на його іконку. Скрипт виконує дії з файлами або папками.

Так само як і аплет, дроплет можна зберігати як змінюваний файл або як виконуваний файл. Змінюваний дроплет містить текст скрипту, тому його можна редагувати та знову компілювати. Щоб редагувати або змінювати дроплет, перетягніть його на іконку Редактора AppleScript. Коли скрипт зберігається як виконуваний файл, редагована частина тексту вилучається. Відтак файл більше не можна редагувати або знову компілювати.

Дроплети вимагають наявності обробника on open, який починає виконувати скрипт під час відкривання дроплета. Обробник on open має параметр, що містить список перетягнутих до дроплета елементів. Частина скрипту дроплета досліджує список, а інша виконує дії з цими елементами. Лише якщо дроплет не розроблено для виконання дії з одним елементом, він має містити цикл, який повторюється відносно елементів списку, виконуючи дії з ними по черзі.

- Нижче наведено приклад обробника дроплета, який використовує команду on open і список параметрів.

```
on open (theItemsDropped)
        --first command
        --second command
end open
```

ПОВ'ЯЗАНІ ТЕМИ

Про аплети
Про скомпільовані скрипти
Типи файлів AppleScript

Основи

Використання словників AppleScript

Щоб знайти команди та об'єкти, які можна використовувати для керування програмою, можна звернутись до словника цієї програми. Словник містить набір термінів, які розуміє програма.

Словник програми можна переглянути в Редакторі AppleScript, вибравши потрібну програму у вікні «Бібліотека» або перетягнувши іконку програми на іконку Редактора AppleScript.

Терміни у словнику об’єднані в групи пов’язаних термінів, які називаються «модулі». Команди та об’єкти виділено жирним шрифтом.

1. У Редакторі AppleScript виберіть меню «Вікно» > «Бібліотека».
2. Двічі клацніть назву програми у вікні «Бібліотека».
3. Виберіть модуль.

   З’являться терміни, використовувані в модулі, та їхній опис.

Мова AppleScript має кілька різних категорій, які наведено нижче.

- Модулі: групи пов’язаних термінів. У словнику Редактора AppleScript модулі показуються у першому стовпчику, позначеному оранжевою буквою «S».
- Команди: потребують дію або результат. Команди, також відомі як «події», показуються у другому стовпчику. Вони позначені синьою буквою «C».
- Класи: об’єкти, що підтримують скриптування.. Опис класу містить список властивостей та елементів цього класу. Класи, позначені бузковою буквою «C», відображаються у другому стовпчику.
- Властивості: об’єкти, що містять значення. Властивості, позначені бузковою буквою «P», відображаються у третьому стовпчику.
- Елементи: об’єкти, що містяться в інших об’єктах. Наприклад, абзац може містити багато слів. Елементи, позначені оранжевою буквою «E», відображаються у третьому стовпчику.

ПОВ’ЯЗАНІ ТЕМИ

Визначення класів
Визначення команд
Використання скриптів AppleScript для керування програмами

ВІДКРИТИ

Помічник із мови AppleScript: Довідка про команди
Помічник із мови AppleScript: Довідка про класи

Основи

Вирішення проблем

Якщо під час використання AppleScript виникла проблема, перегляньте наведені нижче розділи, щоб отримати інформацію, яка допоможе вирішити її.

Якщо виникає помилка під час натиснення користувачем кнопки «Скасувати» у діалоговому вікні «Вибрати зі списку»

Якщо скрипт папки виконується неочікувано

Якщо скрипту не вдається розпізнати диск

Визначення класів

У визначенні класу описано властивості та елементи об’єкта, що підтримує скриптування. Клас можна порівняти з іменником у реченні, а визначення класу — із визначенням цього іменника в словнику.

Зазвичай визначення класу в словнику AppleScript починається з терміну, виділеного жирним шрифтом, і продовжується коротким описом терміну в тому самому рядку.

Потім подається список елементів або об’єктів, які належать скриптованому об’єкту або з яких він складається. Визначення класу не завжди містить список елементів, оскільки деякі об’єкти, що підтримують скриптування, містять інші скриптовані об’єкти, а деякі — ні.

Визначення класу завершується списком властивостей або атрибутів скриптованого об’єкта та їх значеннями.

Запис властивості починається з її назви, після чого подаються такі дані:

- опис формату даних, в якому зберігається значення властивості, наприклад ціле (число), число з десятковою комою (список із двох чисел), рядок (текстові дані) та логічне значення (true чи false)
- необов’язковий індикатор типу [r/o], який показує, що значення властивості придатне «тільки читання» (її можна лише читати, а змінювати не можна)
- короткий опис властивості

Якщо об’єкт указано в множині, він може стосуватися кількох об’єктів.

Якщо об’єкт має список елементів, він містить інші об’єкти. Назву елемента, який можна використовувати у скрипті, виділено жирним шрифтом. Звичайний текст описує спосіб посилання на елемент.

Об’єкти можуть мати властивості. Назви властивостей виділено жирним шрифтом. Звичайний текст показує тип або значення властивості, наприклад, рядок. Властивості можуть бути константами або входити до списку констант. Скрипт може повернути значення властивості та встановити її значення, якщо після синтаксису команди не вказано індикатор [r/o] (тільки читання).

ПОВ’ЯЗАНІ ТЕМИ

Визначення команд
Визначення об’єктів
Команди AppleScript
Використання словників AppleScript

ВІДКРИТИ

Помічник із мови AppleScript: Довідка про класи

Визначення команд

Команди — це терміни мови AppleScript, які виконують дії. Команди показуються у словнику AppleScript в центральному стовпчику. Вони позначені синьою буквою «C». Визначення команди містить короткий опис функцій команди та її синтаксису. Синтаксис — це спосіб написання команди у скрипті. Синтаксис може складатися з кількох частин.

Виділені жирним шрифтом слова — це терміни, які використовуються в операторі. Ці терміни потрібно писати саме так, як написано в словнику.

Синій текст — це надана вами інформація. Наприклад, можна ідентифікувати визначений вами об'єкт або змінну.

Команди іноді мають необов'язкові частини, які пишуться в лапках. Їх можна не використовувати в операторі.

ПОВ'ЯЗАНІ ТЕМИ

Визначення класів
Використання словників AppleScript
Визначення подій
Визначення об'єктів

ВІДКРИТИ

Помічник із мови AppleScript: Довідка про команди

Основні команди

Визначення подій

Визначення події описує дієслово або команду, що використовуються для спрямування дії на керований скриптами об'єкт. Події — це дії або дієслова, які об'єкти можуть виконати або взяти в них участь.

Зазвичай визначення події починається з терміну, після чого у цьому самому рядку подається короткий опис терміну.

Потім наведено список параметрів, які можуть використовуватися під час виконання команди. Необов'язкові параметри взято в лапки, обов'язкові — ні.

Кожний запис параметра починається з назви параметра, продовжується описом формату даних, які використовує параметр, і закінчується коротким описом параметра.

Визначення команди містить короткий опис (підкреслений текст) функцій команди та її синтаксису. Синтаксис — це спосіб написання команди у скрипті. Синтаксис може складатися з кількох частин.

Виділені жирним шрифтом слова — це терміни, які використовуються в операторі. Ці терміни потрібно писати саме так, як написано в словнику.

Звичайним текстом показується надавана вами інформація. Наприклад, можна ідентифікувати визначений вами об'єкт або змінну.

Команди іноді мають необов'язкові частини, які пишуться в лапках. Їх можна не використовувати в операторі.

Команди виконують дії з об'єктами. Визначення об'єкта показує синтаксис для розпізнавання об'єкта, зокрема його назву та короткий опис (підкреслений текст). Клас об'єкта визначає його певні характеристики, наприклад команди, які можна використовувати з цим об'єктом.

ПОВ'ЯЗАНІ ТЕМИ

Події бази даних
Визначення класів
Визначення команд
Визначення об'єктів
Як працює AppleScript

Визначення об’єктів

Команди виконують дії з об’єктами. Визначення об’єкта у словнику AppleScript показує синтаксис для розпізнавання об’єкта, зокрема його назву та короткий опис (підкреслений текст). Клас об’єкта визначає його певні характеристики, наприклад команди, які можна використовувати з цим об’єктом.

Якщо об’єкт указано в множині, він може стосуватися кількох об’єктів.

Якщо об’єкт має список елементів, він містить інші об’єкти. Назву елемента, який можна використовувати у скрипті, виділено жирним шрифтом. Звичайний текст описує спосіб посилання на елемент.

Об’єкти можуть мати властивості. Назви властивостей виділено жирним шрифтом синього кольору. Звичайним синім шрифтом вказано тип значення, яке містить властивість, наприклад тип «string». Властивості можуть бути константами або входити до списку констант. Скрипт може повернути значення властивості та встановити її значення, якщо після синтаксису команди не вказано індикатор [r/o] (тільки читання).

ПОВ’ЯЗАНІ ТЕМИ

Використання скриптів AppleScript для керування програмами
Визначення команд
Визначення класів
Передавання вказівок об’єкту
Використання словників AppleScript

ВІДКРИТИ

Помічник із мови AppleScript: Довідка про класи
Помічник із мови AppleScript: Властивості

Оператори if

Оператор if — це команда, яка виконуються лише за певних умов. Кожен оператор if містить логічний вираз, який оцінюється як істинний або хибний. Команди, які містяться всередині оператора if, виконуються, лише якщо значення логічного виразу є істинним. У кінці оператора if ставиться вираз end if. Наприклад:

```
if amountReceived > 100 then
        set memberStatus to "special"
end if
```

Якщо значення логічного виразу в наведеному вище операторі є істинним (отриманий обсяг більший за 100), виконуються команди всередині оператора if. Якщо значення логічного виразу є хибним, команди не виконуються.

Оператор if може застосовуватися разом з оператором else. Така комбінація відома як оператор if-then-else. Наприклад:

```
if amountReceived > 100 then
        set memberStatus to "special"
else
        set memberStatus to "general"
```

```
end if
```

Якщо значення першого логічного виразу у наведеному вище операторі є істинним (змінна amountReceived більша за 100), виконуються команди першої частини оператора. Якщо значення першого логічного виразу є хибним, тоді виконуються команди другої частини оператора, які розташовані після else.

Можна вкладати оператори if один до одного для оброблення декількох умов.

Докладнішу інформацію про оператори AppleScript можна знайти в посібнику з мови AppleScript.

ПОВ’ЯЗАНІ ТЕМИ

Оператори repeat
Визначення команд
Задання змінних в AppleScript
Значення констант

ВІДКРИТИ

Помічник із мови AppleScript — If

Основні команди

Оператори repeat

Можна використовувати оператор repeat для багаторазового виконання завдань залежно від обставин. Оператор repeat продовжує циклічно виконувати групу операторів, поки не буде досягнуто умови й здійснено вихід із циклу. Можна вкладати оператори repeat один до одного для реагування на умови, що змінюються.

В AppleScript існує низка операторів repeat, які можуть реагувати на різні умови.

repeat (forever)

Циклічне безперервне повторення. Використовуйте оператор exit для виходу з циклу.

```
repeat
        -- statement
        -- statement
        if (x = 5) then exit repeat
end repeat
```

repeat (number) times

Повторення циклу задану кількість разів.

```
repeat 5 times
        -- statement
        -- statement
end repeat
```

repeat while

Повторення циклу, поки виконується визначена умова.

```
set x to 1
repeat while (x < 5)
        -- statement
        -- statement
        set x to (x + 1)
end repeat
```

---

repeat until

Повторення циклу до виконання визначеної умови.

```
set x to 1
repeat until (x = 5)
        -- statement
        -- statement
        set x to (x + 1)
end repeat
```

---

repeat with (variable) from (startValue) to (stopValue)

Установлює значення змінної в початкове значення, потім після кожного циклу збільшує змінну. Оператор repeat завершується, коли змінна досягає кінцевого значення stopValue.

```
repeat with x from (5) to (25)
        -- statement
        -- statement
end repeat
```

---

repeat with (variable) in (list)

Установлює змінну як перший елемент у списку. Після кожного виконання циклу змінна встановлюється на наступний елемент. Оператор repeat закінчується після використання всіх елементів у списку.

```
set myList to {234, 342, 681, 874}
repeat with x in myList
        -- statement
        -- statement
end repeat
```

Докладнішу інформацію про оператори repeat можна знайти в Помічнику з мови AppleScript.

ПОВ'ЯЗАНІ ТЕМИ

Оператори if
Задання змінних в AppleScript
Значення констант

ВІДКРИТИ

Помічник із мови AppleScript — repeat

Значення констант

Константа — це зарезервоване слово в AppleScript, яке містить заздалегідь визначене значення. У мові AppleScript є кілька числових констант, які можна використовувати для отримання інформації щодо об’єктів, здійснювати порівняння або проводити арифметичні обчислення. Програми та інші об’єкти можуть визначати власні константи. Ці константи визначаються у визначеннях класу.

Щоб побачити значення константи, скопіюйте цей приклад до Редактора AppleScript, після чого натисніть «Виконати». Натисніть «Результат», щоб переглянути значення константи.

Результат теж є константою. Це константа скрипту, яка посилається на значення, повернене командою.

```
get the current date
```

До констант належать такі вирази:

- арифметичні константи: pi;
- логічні значення: true, false;
- константи дати та часу: minutes, hours, days, weeks;
- константи рядка: return, space, tab;
- константи скрипту: anything, current application, it, me, missing, value, my, result.

Докладнішу інформацію про константи в AppleScript можна знайти в Помічнику з мови AppleScript.

ПОВ’ЯЗАНІ ТЕМИ

Задання змінних в AppleScript
Визначення об’єктів
Визначення класів
Визначення подій

ВІДКРИТИ

Помічник із мови AppleScript — constants

Задання змінних в AppleScript

Змінна — це контейнер, створений у межах скрипту, який використовується для зберігання значення. Під час створення змінної для неї визначається назва, яка також називається ідентифікатором. Якщо скрипт запущено, значення можна помістити у змінну, отримати доступ до значення або змінити його. Змінні діють тільки під час роботи запущеного скрипту. Якщо скрипт закривається, значення втрачається.

- Ці приклади демонструють, як визначити унікальний ідентифікатор і призначити йому значення.

```
set myVar to "153"

set x to "Квартальні звіти"
```

```
set x to false
```

Щоб побачити, як значення зберігаються у змінних, потрібно вставити ці приклади в Редактор AppleScript і потім натиснути кнопку «Виконати». Натисніть «Результат», щоб переглянути значення, які зберігаються у змінній.

Докладнішу інформацію про змінні в AppleScript можна знайти в Помічнику з мови AppleScript.

ПОВ’ЯЗАНІ ТЕМИ

Значення констант
Оператори if
Оператори repeat

ВІДКРИТИ

Помічник із мови AppleScript — змінні

Основні команди

Визначення розташування програми

Програми в операційній системі OS X існують у вигляді пакетів. Пакет — це спеціальний каталог, який містить папки та файли, що є основою програми. Унаслідок цього команди path to і info for можуть повертати для програми різні результати.

Якщо для програми використовується команда path to, то шлях зазвичай закінчується розширенням .app. Розширення файлів можуть не відображатися у Finder як частина імені файла, але їх все одно можна використовувати для визначення файла у скрипті.

Якщо для отримання інформації про програму використовується команда info for, властивість результату папки зазвичай має значення true. Для отримання додаткових відомостей про файл і його ім’я можна використати додаткові властивості результату, наприклад, приховати розширення, розширення імені або відображуване ім’я.

ПОВ’ЯЗАНІ ТЕМИ

path to
info for
Посилання на папки, файли чи диски
choose application
Команди скриптування Finder

Команди Finder

Посилання на папки, файли чи диски

AppleScript дає можливість посилатися на певні папки, файли і диски за назвами без використання шляху. AppleScript може визначати розміщення таких елементів, як папка «Документи», робочий стіл, домашня папка і стартовий диск.

- У цьому прикладі показано, як можна посилатися на певну папку у скрипті за допомогою її назви.

```
set x to the path to the desktop folder
```

ПОВ’ЯЗАНІ ТЕМИ

Посилання на Finder у скриптах
path to
info for
Команди скриптування Finder

Посилання на Finder у скриптах

Щоб скористатися командами Finder, слід спрямувати їх до Finder в операторі tell так само, як і під час спрямування команд до іншої програми.

- Нижче наведено приклад використання оператора tell у скрипті. Оператор tell скеровує функцію number of windows до Finder.app.

```
tell application "Finder.app"
        return the number of windows
end tell
```

ПОВ’ЯЗАНІ ТЕМИ

Finder
Передавання вказівок об’єкту
Посилання на папки, файли чи диски
Визначення об’єктів
Команди скриптування Finder

ВІДКРИТИ

Помічник із мови AppleScript — специфікатори об’єкта

Команди скриптування Finder

Нижче наведено деякі команди, які можна використовувати у меню Finder:

activate

alias file

alias list

application file

clean up

close

column

Computer-object

container

copy

count

data size

delete

desktop window

desktop

disk

document file

duplicate

eject

empty trash

exists

file

file information

find

Finder

Finder window

folder

get

icon view options

information window

Internet location file

item

list view options

make

move

moving folder window for

open

package

path to

preferences

preferences window

print

quit

Посилання на папки, файли чи диски

Посилання на Finder у скриптах

reveal

select

set

sort

trash

update

window

Стандартні команди додатків скриптування

Додатки скриптування додають багато корисних команд, які можна використовувати у своїх скриптах.

adding folder items to

beep

choose application

choose color

choose file

choose file name

choose folder

choose from list

choose URL

clipboard info

close access

closing folder window for

current date

delay

dialog reply

display

display dialog

do shell script

download

FTP item

get eof

info for

list disks

list folder

load script

mount volume

offset

open for access

open location

opening folder

path to

POSIX path

random number

read

round

run script

say

scripting components

set eof

set the clipboard to

set volume

store script

summarize

system attribute

time to GMT

upload

URL

write